

CLAIMS

1. A multiple-precision, multiply-and-add operation for handling at least one operand having more than one natural word comprising:
 - 5 a first operand;
 - a second operand;
 - an addend operand;
 - a result vector; and
 - for each natural word of the second operand,
 - 10 a block of *multiply-and-add* instructions that multiply the natural word of the second operand by all natural words of the first operand and store results of the multiply-and-add instructions as intermediate results, the block of *multiply-and-add* instructions that multiply the first natural word of the second operand by all natural words of the first operand additionally adding a number of initial natural words of the addend operand to the products of
 - 15 the first natural word of the second operand and all natural words of the first operand, the block of *multiply-and-add* instructions containing no write dependencies.
2. The multiple-precision, multiply-and-add operation of claim 1 wherein each block of *multiply-and-add* instructions contains only *multiply-and-add* instructions.
 - 20
3. The multiple-precision, multiply-and-add operation of claim 1 wherein a block of *multiply-and-add* instructions may contain *add* instructions in addition to *multiply-and-add* instructions.
- 25 4. The multiple-precision, multiply-and-add operation of claim 1 further including:
 - a number of blocks of *add* instructions that add the intermediate results and any remaining natural words of the addend operand to produce a final result vector that contains a sum of the addend operand and a product of the first and second operands.

5. The multiple-precision, multiply-and-add operation of claim 1 wherein at least one of the first operand, second operand, and addend operand is contained within two or more registers.

5 6. The multiple-precision, multiply-and-add operation of claim 1 wherein at least one of the first operand, second operand, and addend operand is contained within two or more natural words in memory.

7. The multiple-precision, multiply-and-add operation of claim 1 wherein the result
10 vector is contained within two or more registers.

8. The multiple-precision, multiply-and-add operation of claim 1 wherein the result vector is contained within two or more natural words in memory.

15 9. The multiple-precision, multiply-and-add operation of claim 1 wherein, because there are no write dependencies in the blocks of *multiply-and-add* instructions, all *multiply-and-add* instructions of each block can be executed together in parallel.

10. A method for multiplying a first operand by a second operand to produce an
20 intermediate product to which an addend operand is added to produce a result in a result vector, at least one of the first operand, second operand, and addend operand having more than one natural word, the method comprising:

for each natural word of the second operand,

using a block of *multiply-and-add* instructions to multiply the natural word of
25 the second operand by all natural words of the first operand and store results of the *multiply-and-add* instructions as intermediate results, when multiplying the first natural word of the second operand by all natural words of the first operand additionally adding a number of initial natural words of the addend operand to the products of the first natural word of the second operand and all natural words of the first operand, the block of *multiply-and-add*
30 instructions containing no write dependencies.

11. The method of claim 10 wherein each block of *multiply-and-add* instructions contains only *multiply-and-add* instructions.
12. The method of claim 10 wherein a block of *multiply-and-add* instructions may
5 contain *add* instructions in addition to *multiply-and-add* instructions.
13. The method of claim 10 further including:
using a number of blocks of *add* instructions that add the intermediate results and any
remaining natural words of the addend operand to produce a final result vector that contains a
10 sum of the addend operand and a product of the first and second operands.
14. The method of claim 10 wherein at least one of the first operand, second operand, and
addend operand is contained within two or more registers.
- 15 15. The method of claim 10 wherein at least one of the first operand, second operand, and
addend operand is contained within two or more natural words in memory.
16. The method of claim 10 wherein the result vector is contained within two or more
20 registers.
17. The method of claim 10 wherein the result vector is contained within two or more
natural words in memory.
18. The method of claim 10 further including executing some or all of the *multiply-and-
25 add* instructions of each block of *multiply-and-add* instructions in parallel.
19. A multiple-precision, multiply-and-add operation for handling at least one operand
having more than one natural word comprising:
a first operand;
30 a second operand;
an addend operand;

for each natural word of the second operand,

- a means for multiplying the natural word of the second operand by all natural words of the first operand and storing results as intermediate results, the means for multiplying the natural word of the second operand by all natural words of the first operand
- 5 additionally adds a number of initial natural words of the addend operand to the products of the first natural word of the second operand and all natural words of the first operand without write dependencies.